



What Exactly do Students Learn when they Practice Equation Solving? Refining Knowledge Components with the Additive Factors Model

Yanjin Long, Kenneth Holstein, and Vincent Aleven
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh PA, 15213
grace.long1030@gmail.com, {kjholste, aleven}@cs.cmu.edu

ABSTRACT

Accurately modeling individual students' knowledge growth is important in many applications of learning analytics. A key step is to decompose the knowledge targeted in the instruction into detailed knowledge components (KCs). We search for an accurate KC model for basic equation solving skills, using data from an intelligent tutoring system (ITS), *Lynnette*. Key criteria are data fit and predictive accuracy based on a standard logistic model called the Additive Factors Model (AFM). We focus on three difficulty factors for equation solving: understanding of variables, the negative sign, and the complexity of the equation. Fine-grained KC models were found to have greater fit and predictive accuracy than an "ideal," more abstract model, indicating that there is substantial under-generalization in students' equation-solving skill related to all three difficulty factors. The work enhances scientific understanding of the challenges students face in learning equation solving. It illustrates how learning analytics could inform the improvement of technology-enhanced learning environments.

CCS CONCEPTS

• **Applied computing**~Interactive Learning Environments • **Applied computing**~Computer-assisted instruction

KEYWORDS

Educational data mining, intelligent tutoring systems, student modeling, K-12, equation solving, knowledge components

ACM Reference Format:

Y. Long, K. Holstein, and V. Aleven. 2018. What Exactly Do Students Learn When They Practice Equation Solving? Refining Knowledge Components with the Additive Factors Model. In LAK'18: International Conference on Learning Analytics and Knowledge, March 7–9, 2018, Sydney, NSW, Australia. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3170358.3170411>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee (Permissions@acm.org).

LAK'18, March 7–9, 2018, Sydney, NSW, Australia

© 2018 Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-6400-3/18/03...\$15.00

<https://doi.org/10.1145/3170358.3170411>

1 INTRODUCTION

Analytics that accurately capture students' knowledge growth during learning activities have many potential applications. They can be used in awareness tools for teachers or students, to visualize how far a particular student or class of students has come in mastering targeted knowledge objectives [41]. They can be used in orchestration tools to help teachers orchestrate personalized classrooms [13,33], or in adaptive learning technologies to adapt the instruction to individual students' learning trajectories [8,9]. Finally, investigations into knowledge growth analytics may help advance the scientific understanding of learning in a particular domain.

The research fields of ITS and educational data mining (EDM) have studied extensively how to assess the knowledge growth of students working with educational software [4,10,35]. This line of work has created models that can accurately predict student performance over repeated opportunities to practice targeted learning objectives, such as the Additive Factors Model (AFM) [7,36] and Bayesian Knowledge Tracing (BKT) [8,9].

Accurate modeling of knowledge growth often requires breaking down the targeted knowledge or skill into smaller elements [20]. In the current paper, we follow a widely-used approach in which knowledge in a given task domain is viewed as comprising a large set of fine-grained knowledge components (KCs) [1,17]. KCs are small "atoms" of knowledge, across which there is assumed to be no transfer. That is, each KC needs to be learned and strengthened through practice and practice with one KC does not strengthen another. For a KC model to be the basis for accurate modeling of knowledge growth, it must identify all KCs that contribute to the given overall skill, including ones that are not well-known or taught directly [18]. Further, it must capture KCs at an appropriate level of generality, in accordance with how students *actually* learn. However, KCs are not directly observable, which makes it challenging to build accurate KC models. Cognitive task analysis has been widely used as an effective technique for manually crafting KC models [29]. For example, in theoretical task analysis, formal models of task structure may be used to infer what KCs are needed to perform the task. Also, in one widely used method for empirical task analysis [29], transcripts of an expert's thinking aloud while performing the task are analyzed to identify KCs. However, inferring KCs from think-aloud data is not straightforward, due largely to the fact that procedural knowledge is non-verbalizable

[3]. While cognitive task analysis can be effective, it is becoming increasingly clear that KC models created through cognitive task analysis methods are not always optimal [1]. Techniques from the field of educational data mining (EDM) for automated or semi-automated KC model discovery and refinement can be a useful supplement. One study combined EDM methods such as AFM modeling and “learning curves” [16] to find a better KC model for an ITS for geometry learning [37]. They then updated the tutor based on the refined KC model and found improved learning outcomes and efficiency due to the redesign [19]. Similarly, learning curves extracted from log data from a tutor for learning SQL were used to explore model improvements [31].

The refinement of a KC model may not only help in improving the educational technology that generated the data, it may also advance our understanding of domain-level learning. In the current work, we focus on Algebra, a challenging domain for many students in US middle schools and high schools. Algebra has been recognized as the “gatekeeper” for learning mathematics [15] and is highlighted by the Common Core Standards, an educational initiative that details learning objectives for K-12 students in the United States. Learning and teaching algebra have been extensively studied in the mathematics education literature [5,15]. However, not much work in the fields of LA and EDM has focused on linear equation solving. It is therefore an interesting open question whether techniques from these fields could extend findings in mathematics education research. More generally, we do not know what KC models accurately model early algebra learning.

In the current work, we systematically search for an accurate KC model for early algebra learning from problem-solving practice with an intelligent tutoring system (ITS) [21]. Specifically, we apply AFM-guided data-driven KC modeling, a proven EDM method, to refine a human-generated KC model for basic algebraic equation solving. We do so using log data from *Lynnette*, an ITS for linear equation solving [26-28,39]. These data come from a classroom study [26] in which working with *Lynnette* for 5 class periods significantly improved students’ equation-solving ability with a large effect size ($d = 0.85$), as measured by a paper and pencil test of equation solving. Thus, we analyze data from students who are learning effectively, by at least one conventional measure. We explore a space of KC models characterized by three difficulty factors in equation solving that are known from the mathematics education literature (e.g., [15]), namely, whether operations involve a variable or a negative sign, and the complexity of the equation. In essence, our analysis asks whether students learn to abstract with respect to these difficulty factors, that is, whether they learn to “see” commonalities between variables and constants, between negative and positive terms, and between more complex versus simpler equations.

The paper is structured as follows: As background, we briefly describe the AFM model and *Lynnette*. We then describe a KC model for early equation solving, formed through theoretical task analysis, that formed the starting point for our data-driven KC model refinement process. We describe how we iteratively refined the abstract KC model based on the three difficulty

factors and present results in terms of fit to the data and cross-validation accuracy of the refined models. We discuss the most accurate KC model produced by our model search process, insights about student learning of early equation solving, and design implications for learning analytics tools.

2 DATA-DRIVEN KC MODEL REFINEMENT USING THE ADDITIVE FACTORS MODEL

There are many ways in which a KC model can be used to make predictions about student performance. The accuracy of these predictions, in turn, can function as a measure of the accuracy with which a KC model characterizes the knowledge students acquire through particular educational activities [1,18-20], an approach we follow in the current work. Specifically, we use the Additive Factors Model (AFM) [7,36], a logistic model that captures students’ knowledge growth as change in their performance over repeated opportunities to practice targeted KCs. The AFM is based on models used in Item Response Theory [40] but extends them in that it assumes an underlying KC model and accounts for the learning that occurs over multiple opportunities to apply each KC. Specifically, AFM predicts the log odds of correct performance on a problem step as a linear function of the student’s initial proficiency, the initial ease of the KCs involved in the step, and the effect of prior practice by the given student on the given KCs (see Fig 1).

$$\log \frac{p_{ij}}{1-p_{ij}} = \theta_i + \sum_k \beta_k Q_{kj} + \sum_k Q_{kj} \gamma_k T_{ijk}$$

Given variables:

p_{ij} (0 or 1) probability that student i gets step j correct

Q_{kj} (0 or 1) whether KC k is needed for step j

T_{ijk} number of opportunities student i has had to practice KC k , prior to step j

Estimated parameters:

θ_i proficiency of student i

β_k ease of KC k

γ_k gain for each opportunity to practice KC k (learning rate)

Figure 1: The Additive Factor Model for using a KC model to make predictions about student performance

AFM assumes that each student starts with her own initial proficiency level, which affects her performance across KCs and opportunities. Further, AFM assumes that different students learn the same amount from each practice opportunity for any given KC. AFM assumes further that KCs differ both in their initial ease and in the amount of learning that occurs on each opportunity. Depending on the parameter values, AFM predicts a gradual increase of expected performance, with smaller returns of later practice, asymptoting at perfect performance. Application of AFM requires a KC model, including a mapping of the steps in the practiced problems to KCs. This mapping is captured by Q_{kj} in the formula above. AFMs are typically

estimated using tutor log data, which ITSs by now generate routinely.

AFMs make several simplifying assumptions about student knowledge growth. For example, a limitation of standard AFM is that, apart from each student's initial proficiency, it does not capture possible individual differences, such as student-specific learning rates or student-specific KC ease parameters. Another limitation is that AFM assumes that all practice opportunities for a given KC influence learning in the same way, regardless of what happened on each opportunity (e.g., whether the student answered correctly or how much time she spent). Despite these limitations, AFM has turned out to be very useful in practice, for example in helping to refine KC models in given task domains [1,18,19,23], as we do in the current paper. It may be interesting to mention that recent EDM work has started to address some of the limitations of the standard AFM model, for example, by introducing student-group learning rates [25], taking into account the correctness of past performance [34] or students' help requests [12], and weighing recent performance more heavily than more distant past opportunities [11].

3 INTELLIGENT TUTORING SYSTEM USED

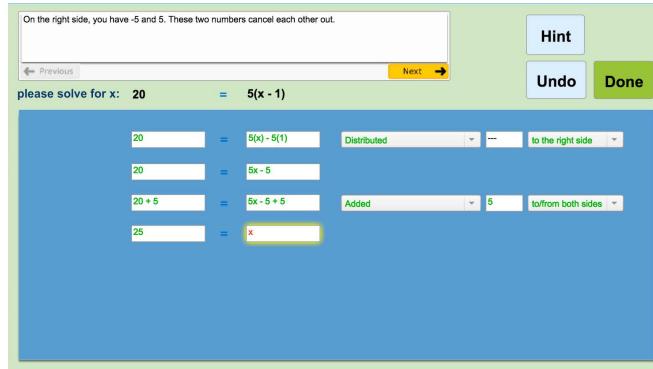


Figure 2. The problem-solving interface of *Lynnette*.

As mentioned, in the current study we use a data set from an ITS for basic equation solving called *Lynnette* (see Fig. 2) [26–28,39]. *Lynnette* is implemented as a rule-based Cognitive Tutor using the Cognitive Tutor Authoring Tools (CTAT, [2]). Using *Lynnette*, students solve equations step-by-step, with detailed step-level guidance from the tutor in the form of hints, correctness feedback, and error-specific feedback messages. Within problems, *Lynnette* recognizes major and minor strategy variants and follows along with the strategy the student selects. *Lynnette* offers practice with five levels of equations with increasing difficulty (see Table 1). Each level contains variations of the example equations shown in the Table (e.g., negative variable coefficients, negative constants, negative solutions, different orders of variables and constants, left and right side swapped, etc.). Within each level, *Lynnette* supports individualized mastery learning based on Bayesian Knowledge Tracing, a commonly used algorithm for tracking skill mastery based on students' interactions with educational software [8,9].

Table 1. Types of Equations Practiced in *Lynnette*

Levels in the Tutor	Example Equation
Level 1	$-x + 6 = 3$
Level 2	$2x + 3 = 9$
Level 3	$2x + 6 = 5x$
Level 4	$2x + 4 = 7x - 6$
Level 5	$2(x + 1) + 3 = 13$

Like many other ITSs, *Lynnette* generates detailed log data that captures students' transactions with the software. These data include students' correct and incorrect attempts at solving problem steps and their requests for on-demand hints, the tutor's response (i.e., its assessment of whether the steps were correct, and its determination of which KCs were involved, based on its KC model, as well as any hints given). We leveraged these log data in our search for an accurate KC model, described below.

4 KC MODELS EXPLORED

In our KC model search, we worked from more general KC models to more specific models. The initial model was created through theoretical cognitive task analysis. The more specific models were created by “splitting” KCs in the more general models, based on our three difficulty factors. Our search strategy was geared towards discovering the cumulative effect of splits based on these difficulty factors. The splitting was achieved by relabeling steps in the *Lynnette* log data based on finer-grained KC representations, thereby changing the step-skill mapping encoded by Q_{kj} in the AFM model (see Figure 1). In other words, when a KC was split into, say, two separate KCs, steps in the log data originally labeled with that KC would now be labeled with one of the two new KCs.

4.1 The Fully Abstract Model

Table 2. Fully Abstract KC Model for Equation Solving (FAME)

KCs in the FAME model	Levels
Add/subtract simple term	L1 – L5
Divide both sides by variable coefficient	L1 – L5
Combine like terms	L1 – L5
Divide simple terms	L2 – L5
Distribute product	L5
Multiply simple terms	L5

As a starting point for our search, we use a KC model for basic equation solving that we created through theoretical cognitive task analysis. In this theoretical task analysis, we identified the main transformations and simplifications that can be applied when solving basic equations, and assumed a single KC for each. We call the resulting KC model the Fully Abstract Model for Equation-solving (FAME). This model (shown in Table 2) breaks down the overall skill of solving basic equations into 6 KCs. Two KCs model two main transformations in equation solving: add/subtract the same term to/from both sides, and divide both sides by the same term. Four KCs model four types of

simplifications: combining like terms, distributing a product across a complex term, and multiplying or dividing simple terms. Arguably, the 6-KC FAME model represents “ideally abstracted” knowledge of basic equation solving. Under this model, a student (in their problem-solving behavior) “sees” commonalities between constants and variables, between positive and negative terms, and between simpler and more complex equations.

We then created more specific models, by splitting KCs in the FAME model based on our three difficulty factors.

4.2 Understanding of Variables

Our first difficulty factor is the handling of variable terms when solving equations. Prior mathematics education research shows that it is difficult for students to develop a correct understanding of the concept of variables [15]. We have also observed informally in our prior classroom studies with *Lynnette* that students process the same transformation differently when a variable is involved instead of a constant. Although variables have been recognized as a challenging concept in early algebra learning, we are not aware of any investigations of how this difficulty affects the learning of equation-solving skill and whether students, as they learn to transform and simplify equations, abstract each operation across constant terms and variable terms. Therefore, we investigated KC models that split each operation that can be applied to both variables and constants into separate KCs, one KC for the given operation applied to a variable term and one KC for the given operation applied to a constant term. Three of the 6 KCs in the Fully Abstract Model were split in this manner, namely, those for adding/subtracting a term from both sides, combining like terms, and computing a quotient computing a quotient. The resulting model has 9 KCs. We call it the FAME-C/V model (see Table 3).

Table 3. The FAME-C/V model

KC	Levels	Abbrev.
Add/subtract constant	L1 – L5	AC
Add/subtract variable	L3, L4	AV
Divide both sides by variable coefficient	L1 – L5	DV
Combine constant terms	L1, L2, L4, L5	CC
Combine variable terms	L3, L4	CV
Compute quotient for constant	L2 – L5	CQC
Compute quotient for variable coefficient	L2 – L5	CQV
Distribute	L5	DIS
Multiply simple terms	L5	MST

4.3 The Negative Sign

The negative sign, our second difficulty factor, has been documented in prior literature as posing challenges in learning equation solving. Booth and Koedinger [5] point out that students often fail to recognize the negative sign as part of a term of an equation. For example, in equation $4x - 3 = 9$, students have trouble recognizing “ $- 3$ ” as a term, which results in errors such as subtracting “ 3 ” instead of “ $- 3$ ” from both sides.

Although the negative sign has been recognized as a difficulty factor, we are not aware of any investigations of how it affects the learning of equation solving, or of whether students learn operations that abstract across positive and negative variable terms, as well as across positive and negative constant terms. Further, we must consider the possibility that students may treat “ $- x$ ” as a special case of negative variable terms, as was found in some prior work with simulated (but not “real”) students [22]. We therefore investigated whether negative variable terms (i.e., terms with an explicit negative coefficient, such as “ $- 2x$ ”), as well as negative x (i.e., “ $- x$ ”), lead to greater difficulty in students’ equation solving, compared to positive variable terms. Specifically, we focused on three KCs in the FAME-C/V model that often operate on variable terms, namely, add/subtract constant (AC), add/subtract variable (AV) and divide both sides by variable coefficient (DV), to investigate how the presence of the negative sign (namely, negative variables, negative x , and negative constants), influences students’ learning these KCs. We apply these splits to the FAME-C/V model because in our model search, we found that the FAME-C/V model is more accurate than the FAME model. We first look at three models that apply each split separately and then at a model that applies all three splits simultaneously. We call the latter model the FAME-C/V-Neg model. This model has 14 KCs.

4.4 Complexity of the Equation

As a third difficulty factor, we explored whether complexity of the equation influences difficulty and whether students, as they learn to solve equations, learn abstract operations that apply across simpler *and* more complex equations, or learn specific operations that do not. On the one hand, a more complex equation may make solving steps harder, because there are more distractors – more choices to make regarding how to apply an operation and therefore more opportunity for error. For example, the error of combining unlike terms is not a possibility in the simplest equation types, where unlike terms never occur on the same side of the equation; similar observations may be true for other types of errors and other KCs. On the other hand, by the time students get to more complex equations (which in *Lynnette* occur later on, see Table 1), they may have mastered many KCs, which may have the opposite effect. Thus it is somewhat difficult to predict the “net effect” of the complexity difficulty factor.

Because the Levels in *Lynnette* address progressively more complex equations (as shown in Table 1 above), we use the Level as indicator of equation complexity on which to base the splits. In our model search process, the split based on level is applied to the FAME-C/V-Neg model, because this model turned out to be the most accurate model based on previous splits. We split each KC in this model, to produce a separate KC for each level on which the original KC occurs (see Table 3). The resulting model (the Fully Specific Model) has 41 KCs. With splits based on all three difficulty factors, it is the most specific model explored in our model search process.

5 METHODS

Our main goal was to explore whether a more accurate KC model for equation solving could be found by accounting for the three difficulty factors just described: understanding of variables, the negative sign, and the complexity of the equation.

5.1 Dataset

We used a dataset from a classroom study with 267 participating 7th and 8th grade students [26], in which *Lynnette* was used by a total of 215 students (i.e., who were present for at least one day of the study). Table 4 provides descriptive statistics for this dataset. The dataset has 197,233 transactions between the students and the tutor, which were recorded in DataShop (a major educational data repository) [16]. The students practiced solving linear equations with *Lynnette* from 1 to 5 class periods, with an average of 136.8 minutes (SD=37.04) per student spent in the tutor. The students had received only a limited amount of classroom instruction and practice on equation solving before the study. Thus, the data reflects early equation-solving practice. Results from paper pre- and post-tests revealed that the students improved significantly on solving equations practiced in the tutor [26], indicating that the transaction data reflects student learning that by standard measures should be considered effective and successful.

Table 4. Descriptive statistics for DataShop dataset.

Number of Students	215
Number of Unique Steps	10,961
Total Number of Steps	95,419
Total Number of Transactions	197,233
Total Student Hours	488.92

5.1 Model Search Procedure

We used the FAME model as the basis for model search and evaluation. We explored refinements of this model based on the three difficulty factors. Specifically, we repeatedly tested whether a KC model that is specialized based on one of the difficulty factors is more accurate than a model not specialized in that manner but otherwise the same. In the next round, we kept the more accurate of the two models, and specialize this more accurate model based on the next difficulty factor. This way, we study the cumulative effect of the three difficulty factors.

In each step of the model search process, we first analyzed how the given difficulty factor affects students' performance of equation-solving steps with the tutoring software. Specifically, we compared students' number of errors on steps with the given difficulty factor versus corresponding steps without that factor. This analysis serves as a first verification that the hypothesized difficulty factor is indeed a difficulty factor. Next, we ran the AFM analysis for the fine-grained KC model implied by the new difficulty factor. Specifically, we investigated whether the splits based on the difficulty factor (applied to the best model so far) lead to better model fits (using AIC) and greater predictive accuracy in cross-validation. The analyses based on error rate

and on AFM are complementary, because a difference in one measure does not necessarily imply a difference in the other. For example, the error rate captures how the student performed over a range of practice opportunities in the aggregate; in addition, by comparing performance on steps with and without a difficulty factor, it shows *how* a difficulty factor affects the difficulty of problem-solving steps. By contrast, the AFM analysis measures *learning*; it accounts for changes over time (in the learning rates).

We note that for the first type of analysis in our model search (i.e., comparison of error rates) the results may be skewed by differences in the number of practice opportunities across KCs. For example, as we will see below, there were more instances of operations applied to constant terms than to variable terms. Thus, observed differences in student error rates on steps involving constant versus variable terms may be accounted for, at least in part, by the fact that students had more opportunities to learn operations on constants, compared to variables. Such differences however do not play as large a role in the AFM analysis, because this analysis considers the order of the practice opportunities.

6 RESULTS

For each step in our model search we first present the results of analyzing students' performance on the tutor problems, namely, students' incorrect attempts on problem steps that practice the targeted KCs. Second, we report the results of AFM analysis run in DataShop [16] in terms of the AIC (Akaike Information Criterion) and the root mean-squared error (RMSE, unstratified) for cross validation when running AFMs [38]. A decrease of 10 points (or more) in AIC was deemed to be a reasonable indicator of better model fit according to prior practice [6]. In combination with decreases in AIC, a relatively small decrease in RMSE was considered further evidence of better model fit, given prior evidence that even seemingly small changes in RMSE across KC models can correspond to large instructional benefits (e.g., when these KC models are used to drive adaptive problem selection in an ITS) [24,42].

6.1 Variable versus Constant

Table 5. Errors per Step for Corresponding Variable and Constant KCs

KC	Constant		Variable	
	Steps	Errors	Steps	Errors
Add/Subtract	3836	0.63 (1.68)	2603	0.77 (1.28)
Combine Like Terms	8995	0.41 (1.04)	5741	0.53 (1.09)
Compute Quotient	5031	0.34 (0.90)	3861	0.16 (0.56)

We started by investigating the difficulty factor: understanding of variables. First, we compared students' performance with the tutor on steps with KCs that capture operations on variable terms versus the corresponding operation applied to constant

terms. As shown in Table 5, when adding or subtracting to transform the equation or when combining like terms to simplify the equation, students made more incorrect attempts on steps with variable terms (AV, CV) than on steps with constant terms (AC, CC). On the other hand, computing a quotient was easier for a variable term (CQV) than a constant term (CQC). One reason for the latter finding could be that the steps that practice CQV were (by definition) steps where students were simplifying ax/a to x , where the divisor is always chosen to be the same as the variable coefficient, so as to isolate x . By contrast, for CQC (i.e., for constants), the steps typically involve computing more complicated (integer) quotients.

Table 6. Model fit and predictive accuracy for each refinement in our model search.

Model	KCs	AIC	RMSE
Fully Abstract Model (FAME)	6	40,723.72	0.411159
FAME-C/V	9	39,630.28	0.404570
FAME-C/V-AC	10	39,538.68	0.403559
FAME-C/V-AV	11	39,535.27	0.403944
FAME-C/V-DV	11	39,493.90	0.403882
FAME-C/V-Neg	14	39,302.37	0.402676
Fully Specific Model	41	38,596.16	0.402052

To confirm that students process variables and constants differently, we compared AFM model fits and predictive accuracy for the Fully Abstract Model (FAME) against the FAME-C/V model, which as discussed has splits based on whether operations involved constants or variables. As shown in Table 6 (row 2), the more specific model (FAME-C/V) has better fit (lower AIC) and predictive accuracy (lower RMSE).

The effect of splitting a KC can be observed by inspecting “learning curves” [1,20] extracted from the tutor log data (see Figure 2; the learning curves shown here were generated using DataShop [16]). In general, learning curves display the student error rate over successive opportunities to apply a given KC, averaged over students. The red lines show the data, the blue lines predictions based on AFM, with parameter values obtained by fitting AFM to the data under the given KC model. In Figure 2 we see learning curves for the KC for adding/subtracting a term from both sides (top), and the two KCs into which it is split, one for adding/subtracting a constant term (AC, left), one for adding/subtracting a variable term (AV, right). The learning curves show that one of the split KCs (subtracting a variable term) is more difficult than the other (subtracting a constant term). Treating them as a single KC, as FAME does, therefore does not lead to good fit with the data. (The fit can be assessed visually by how closely the blue line tracks the corresponding red line.) The fact that the predicted line for the abstract KC (Figure 2, top) slopes upward is also suggestive of KC model misspecification [1], although other factors may be at work as well (e.g., a sampling effect, due to weaker students receiving more practice opportunities, as well as complications arising from the fact that some steps are labeled with multiple KCs).

6.2 Negative Sign: Add/Subtract Constant

Next, we moved on to our second difficulty factor: the negative sign. As mentioned, we focused on three KCs that might be influenced by the presence of the negative sign, add/subtract constant (AC), add/subtract variable (AV) and divide both sides by variable coefficient (DV). We start by looking at each of these KCs separately.

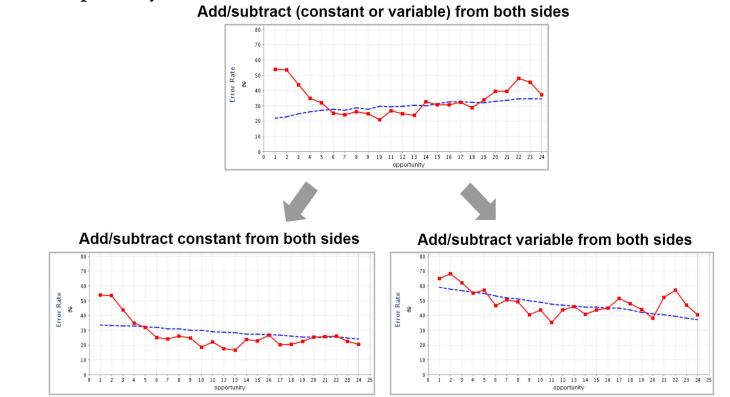


Figure 2. Top: learning curve for adding/subtracting a term from both sides; Bottom: curves for two KCs into which it is split (based on whether the term is a constant or variable).

First, we investigate whether the students made more errors adding or subtracting a constant from both sides of the equation when it was negative, compared to when it was positive. As shown in Table 7, the average number of incorrect attempts per step for adding/subtracting negative constants is slightly higher than adding/subtracting positive constants. However, given that this difference is small and given that there were only a limited number of steps in which the students added or subtracted a negative constant, the analysis of error rates seems inconclusive.

Table 7. Errors Per Step for Adding/Subtracting Positive and Negative Constants

	Positive Constants	Negative Constants
Number of steps	3673	163
Errors (SD)	0.61 (1.68)	0.66 (1.14)

We also studied the same issue by comparing KC models using AFM. We compared the best model from the previous step (FAME-C/V) to one in which the KC for adding/subtracting a constant from both sides (AC) was split based on whether the constant is positive or negative (FAME-C/V-AC). In spite of the small difference in error rate between positive and negative constants, discussed above, the improvement of the more specialized KC model, both in terms of AIC and RMSE, is substantial (see Table 6, row 3), compared to the KC model from which it derives.

6.3 Negative Sign: Add/Subtract Variable

We also looked at how the negative sign influences students' performance on steps that added or subtracted a variable from both sides. We separated the steps based on whether a positive variable, a negative variable or a negative x was added or subtracted on the steps that involve the add/subtract variable (AV) KC. As shown in Table 8, both a negative variable and a negative x seem to make the steps more difficult than a positive variable (although the number of steps differs across the three cases).

Then, we compared FAME-C/V, the best model from the previous step, against a model in which the AV KC was split into three separate KCs, AV-positive, AV-negative and AV-negativeX, creating a new model called FAME-C/V-AV. As shown in Table 6 (row 4), this split improves the accuracy of the model. Both the AIC and RMSE measures show that FAME-C/V-AV is a better model than FAME-C/V.

Table 8. Errors for two KCs split based on the variable term

	Positive Variable		Negative Variable		Negative x	
	Steps	Errors	Steps	Errors	Steps	Errors
Add/ Subtract	1544	0.66 (1.20)	748	0.90 (1.35)	311	1.14 (1.45)
Divide	2012	0.39 (0.84)	1411	0.75 (1.20)	1311	0.79 (1.17)

6.4 Negative Sign: Divide by Variable Coefficient

Lastly, we investigated the influence of the negative sign on dividing both sides by the variable coefficient (DV). Specifically, we compared the average number of incorrect attempts per step for the transformation steps that involve dividing a positive variable, a negative variable, and negative x . As shown in Table 8, the students made almost twice as many errors on the negative variable steps as the positive variable steps. The steps involving negative x were marginally more difficult than other negative variable steps.

We compared FAME-C/V (the best model so far) against one in which DV is split into three new KCs (DV-positive, DV-negative and DV-negativeX). This new model also performs better, both in terms of AIC and RMSE (see Table 6, row 5).

6.5 All Splits Based on the Negative Sign

The results presented in the previous three subsections show that the FAME-C/V model can be improved by splitting three KCs based on whether the terms involved are positive or negative. So far, however, we considered these three splits separately. We now consider a single model containing all three splits, namely, FAME-C/V-Neg. As shown in Table 6 (row 6), this model is more accurate than FAME-C/V both in terms of AIC and RMSE. It is also more accurate than each of the three models that split a single KC based on the negative sign.

6.6 Complexity of the Equation

As our third and final difficulty factor, we consider the complexity of the equation, as indicated by the level in *Lynnette* at which the equation occurs (see Table 1). First, we find that the error rate of each KC in the Fully Abstract Model (FAME) varies by level, as shown in Figure 3, with a pronounced spike in the error rate in between levels 2 and 3, a point to which we return below. Thus, this analysis strongly suggests that the level (as a proxy for complexity) is a difficulty factor not captured in FAME.

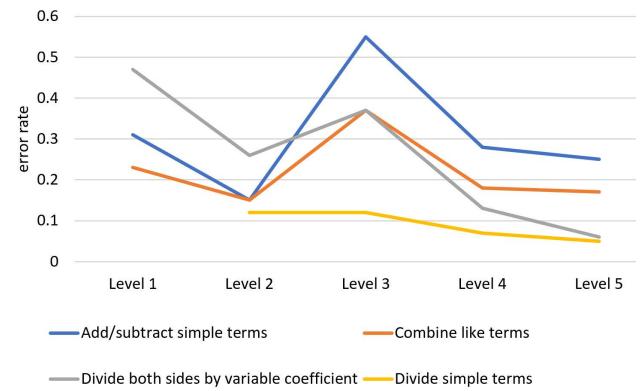


Figure 3. Average error rate, per skill, across levels in Lynnette (for KCs in FAME that appear in multiple levels).

Following our model search strategy, in our AFM analysis we consider this difficulty factor in the context of the best model so far, FAME-C/V-Neg. Thus we compare FAME-C/V-Neg and the Fully Specific Model. We find, once again, that the more specific model has a better fit with the data (see Table 6, row 7). Splitting KCs based on the complexity of the equation yields a more accurate model.

It is interesting to consider *how* the complexity of the equation may influence the difficulty of applying specific KCs. As mentioned, Level 3 has a higher error than the other levels (see Figure 3), suggesting that the transition from regular 2-step equations of the form $ax + b = c$ (practiced at Level 2) to 2-step equations of the form $ax + b = cx$ (practiced at Level 3) is challenging. It may be that practicing $ax + b = c$ on Level 2 led to shallow knowledge such as learning to subtract, from both sides, the constant that occurs on the side where there is also a variable term. This strategy works on Level 2 but fails on Level 3. On Level 3, subtracting the constant b does not bring the student any closer to a solution. *Lynnette* allows this move (i.e., accepts it as correct in spite of it being un-strategic) on the assumption that students could learn from seeing consequences of un-strategic moves. The high error rate on Level 3 however suggests that recovering from this un-strategic move is hard (and that our assumption is not correct). This interpretation is consistent with the fact that the most difficult skill at Level 3 is to add or subtract a term from both sides (see Figure 3). Thus, our complexity difficulty factor appears primarily to reflect a specific difficulty, rather than the general complexity of the equation.

6.7 Final Outcome of the Model Search

At the end of our model search, the Fully Specific Model best accounts for the data (i.e., has both the lowest AIC and RMSE). Figure 4 illustrates that compared to the Fully Abstract Model (FAME, shown on the left), the AFM's predictions based on the Fully Specific Model (on the right) better fit the data. The mismatch of the AFM predictions with FAME's data is indicative of KC model misspecification [1]. It is possible as well that a sampling effect is at play (e.g., a mastery attrition bias, by which lower-achieving students are over-represented on later opportunities, because they need more practice to meet the tutor's mastery criterion) [14].

7 DISCUSSION

The current study demonstrates successful use of a data-driven approach to refine a KC model for equation solving so that it accurately models students' knowledge growth. The work combines findings from domain-specific education research, students' performance data, and AFM modeling to improve a KC model used in an ITS. We found that a specialized KC model for equation solving that accounts for three difficulty factors

adding/subtracting a term from both sides and when dividing both sides by the variable coefficient. Moreover, negative variables and negative x are more difficult than negative constants. This finding suggests that students have difficulties understanding commonalities between positive and negative terms, and learn operations that do not abstract across positive and negative terms. Further, the presence of a negative variable term with implicit coefficient appears to make the adding/subtracting transformation and the division transformation more difficult, compared to working with a variable term with a positive coefficient, or with an explicit negative coefficient. This finding suggests that students have trouble learning operations that abstract across negative variable terms with implicit and explicit negative coefficients. (They may separately learn versions of each operation for positive and negative terms.) Our findings regarding the specific circumstances under which a negative sign makes an equation more difficult and the skill harder to be generalized have not, to the best of our knowledge, been reported in prior math education or EDM research.

Finally, we found that the difficulty of equations varies with their complexity, encoded as the specific level at which it occurs

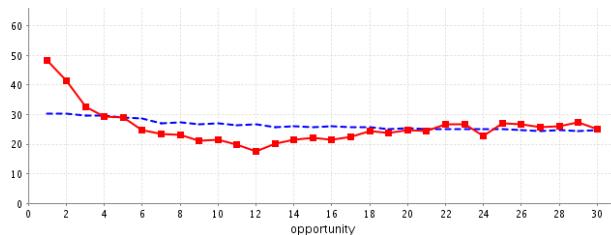
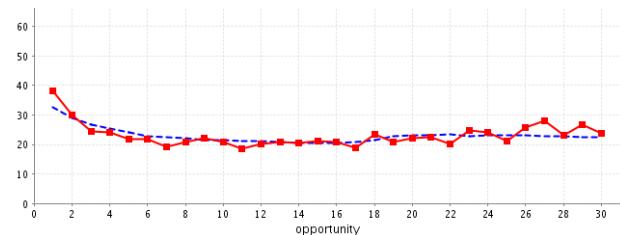


Figure 4. Learning curves for the FAME model (6 KCs) (left) and the Fully Specific Model model (41 KCs) (right). Red lines show the data; blue dotted lines show predicted performance based on AFM, as applied to the given KC model.



(variables, the negative sign, and the complexity of the equation) predicts student performance best. It does so more accurately than a general KC model based on theoretical task analysis, which arguably represents the ideal outcome of learning equation solving.

These findings indicate substantial under-generalization in students' equation-solving knowledge at early stages of learning. First, students appeared to have greater difficulty performing operations on variable terms than on constant terms, specifically, when adding/subtracting a term from both sides and when combining like terms. Apparently, they have trouble "seeing" commonalities between operating on constant terms and operating on variable terms, and may learn specialized versions of each operation that handle only one or the other. By contrast, computing a quotient is easier when a variable term is being divided, compared to dividing a constant term, likely because dividing by the variable coefficient always yields a quotient of 1. This finding, too, may point to under-generalized student knowledge.

Second, we found that operating on negative terms (negative constants, negative variables, and negative x) tends to be more difficult than operating on positive terms, when

within the sequence of levels in *Lynnette*. Thus, students' knowledge tends to be tied to the complexity of the equation. Initially we assumed that the complexity might influence difficulty with two competing effects: more complex equations provide greater opportunity for errors, but are taken on by the student with stronger skills learned on prior levels. Although our analyses do not rule out these effects, unexpectedly, we uncovered a specific effect of equation complexity, namely, that transitioning from $ax + b = c$ to $ax + b = cx$ is challenging. This specific difficulty in early algebra learning, and the challenge of learning general operations that avoid it, appears to be a new finding in the literature on EDM.

It is important to note that the kind of under-generalized knowledge we report can lead to correct problem-solving performance in many instances, for example when students develop mastery of a full set of split KCs, and even when they master only a subset of the split KCs, although then only on problems involving the mastered KCs. The strong pretest to posttest gains on a test of basic equation solving that we saw in the current study [26] attest to this fact. Nonetheless, it is our belief that the kind of under-generalization we find in students' equation-solving skill is undesirable, from an educational point

of view (cf. [32] who considered learning a more abstract KC model as a measure of instructional efficacy). Learning many specialized KCs is probably highly inefficient, compared to learning properly-abstracted knowledge, simply because there are many more KCs to be mastered. Further, although the students in our study made strong gains from pretest to posttest, it is likely that they would have made even stronger gains if they had learned more general KCs. (Our data do not speak to that issue, however.) Further, the observed under-generalization may reflect a lack of conceptual understanding (although our data do not speak directly to this issue either).

One possible interpretation of our results is that learning properly abstracted problem-solving knowledge requires strong conceptual knowledge and that problem-solving practice even when supported by an ITS does not always yield strong conceptual knowledge. An alternative interpretation of these results is that learning under-generalized knowledge early on in the process of acquiring a complex cognitive skill may not in and of itself be cause for concern, if later learning helps students generalize the initially over-specific knowledge. It would be interesting to investigate what developmental paths or learning activities lead from initially under-generalized knowledge to properly-abstracted knowledge.

It is an open question to what range of learning environments the results of our study will generalize. As mentioned, an ITS, the environment with which the results were obtained, provides a high level of guidance. It is an open question whether the results will generalize to environments with a lower level of assistance. Our hunch is that they will; that lowering the level of assistance will not by itself result in more abstract learning. We believe the current results suggest limitations of problem-solving practice, with or without ITS.

The work contributes to KC modeling in algebra; as discussed, more accurate KC models can help in more accurately modeling students' knowledge growth. So far, the fields of educational data mining and learning analytics have not used the advanced methods they have developed specifically to the area of algebra learning, as we do in this paper. Although [19] used AFM to analyze a data set about equation solving, no details were given.

The work also contributes to understanding of challenges that middle school students face in learning equation-solving skill. Although variables and the negative sign had been identified as difficulty factors in earlier literature [5,15], their influence on student learning of equation-solving skill had not been studied. We are not aware of any prior work that studied the influence of equation complexity. This question tends to be hard to tackle without rich data sets and analytic methods such as those used in the current work.

The work confirms the power of AFM-guided KC model refinement [7] even in a highly-studied domain such as algebra. It confirms the notion that without data-driven model refinement, it is difficult to get a KC model right. We note that AFM-guided KC model refinement is not limited to ITS data, even if it has been applied most often to that kind of data. For example, a study by Lovett et al. [30] shows truly dramatic

results of the redesign of an online statistics course based on data-driven KC model refinement. In general, what is needed for this kind of analysis is to define KCs, tag learning activities (or preferably, steps of learning activities) with KCs (this can be done after the fact) and instrument the software to log all student transactions, together with correctness information.

Finally, even though in general, there is no straightforward method to derive instructional design recommendations from a refined KC model, we see many design implications of the current result. First, if (as discussed) a lack of conceptual understanding contributes to the observed under-generalization in students' problem-solving knowledge, then it would be useful to try to design conceptually-oriented instruction that helps students "see" important generalizations, such as commonalities between variable and constant terms, or between positive and negative terms. (Researchers have demonstrated ways of improving conceptual knowledge in algebra, including methods embedded in ITSs [5]; it is an interesting open question whether they lead to more abstract equation-solving knowledge). A different approach would be to adapt the ITS to make problem-solving practice more effective (cf. [19]). A simple approach would be to modify *Lynnette* so it helps students practice the most accurate KC model to mastery (i.e., the Fully Specific Model). It is not clear, however, that it would be worth the time it takes students to practice all 41 specialized KCs to mastery. Alternatively, it would be very interesting to try to devise methods by which an ITS, equipped with different KC models, some more abstract, some more fine-grained, could identify students who seem to be learning more abstractly versus students who have trouble seeing the abstractions, and adjust instruction accordingly on an individual basis. We are not aware of any ITS capable of adapting to student learning in this manner.

The work also has interesting implications for the design of teacher support tools that leverage analytics about students' knowledge growth. For example, it would be interesting to try to devise methods by which a teacher dashboard could report which of several increasingly specific KC models best fits the data for a given class of students. It would be helpful, in addition, to design automatic methods that could identify groups of students within a class who appear to be learning abstract KCs, versus groups who appear to be learning overly-specific KCs. In response to such information, a teacher might provide conceptual instruction focused on helping students see commonalities they are not currently seeing, according these analytics, perhaps even tailoring the extra conceptual instruction to specific groups of students. This approach exemplifies a novel way that analytics could be used to support more personalized classrooms [13,33,41]. Future work should explore these (and other) design implications, and should preferably conduct "close the loop" studies that test whether the present findings could be leveraged to improve algebra instruction [19,24,42]. In addition, we recommend the iterative refinement of KC models as a powerful way in which learning analytics and educational data mining can be used to productively inform instructional design

and the design and iterative improvement of technology-enhanced learning environments.

8 CONCLUSION

The current work illustrates that AFM-guided model refinement search can be effective in identifying a KC model with improved predictive accuracy even in an area as well-studied as basic algebra. Furthermore, the work illustrates that this model search process can advance theoretical understanding of student learning. Our discoveries with respect to the under-generalization in student learning extend prior findings from math education regarding difficulty factors in algebra. The refined model has profound implications for how to design better algebra instruction (cf. [19,24,42]). The work illustrates that EDM methods have much to offer to Learning Analytics, despite these fields' differences in focus.

ACKNOWLEDGMENTS

We thank Abdel-Rahman H. Ahmed and Cindy Tipper for their kind help with this work. We would also like to thank the participating teachers and students. This work was funded by an NSF grant to the Pittsburgh Science of Learning Center (NSF Award SBE083612), and by IES, through Grant R305B150008 to CMU. The opinions expressed are those of the authors and do not represent the views of the NSF, IES, or the U.S. Department of Education.

REFERENCES

- Aleven, V. and Koedinger, K. R. 2013. Knowledge component approaches to learner modeling. *Design recommendations for adaptive intelligent tutoring systems*, I, 165–182, Orlando, FL: US Army Research Laboratory.
- Aleven, V., McLaren, B. M., Sewall, J., & Koedinger, K. R. 2006. The cognitive tutor authoring tools (CTAT): Preliminary evaluation of efficiency gains. In *ITS 2006*. Berlin, Heidelberg: Springer-Verlag.
- Anderson J. R. Rules of the mind. Hillsdale, NJ: Lawrence Erlbaum Associates; 1993.
- Baker, R.S. and Yacef, K. 2009. The state of educational data mining in 2009: A review and future visions. *JEDM* 1, 1, 3-17.
- Booth, J. L., & Koedinger, K. R. 2008. Key misconceptions in algebraic problem solving. In *CogSci*, 571-576. Austin, TX: Cognitive Science Society.
- Burnham, K. P., & Anderson, D. R. (2004). Multimodel inference: understanding AIC and BIC in model selection. *SMR*, 33(2), 261-304.
- Cen, H., Koedinger, K., and Junker, B. 2006. Learning factors analysis - a general method for cognitive model evaluation and improvement. In *ITS 2006*, 164–175, Berlin, Heidelberg: Springer-Verlag.
- Corbett, A.T. and Anderson, J.R. 1995. Knowledge tracing: Modeling the acquisition of procedural knowledge. *UMUAI* 4, 4, 253-278.
- Corbett, A., McLaughlin, M., and Scarpinatto, K.C. 2000. Modeling student knowledge: Cognitive tutors in high school and college. *UMUAI* 10, 81-108.
- Desmarais, M.C. and Baker, R.S. 2012. A review of recent advances in learner and skill modeling in intelligent learning environments. *UMUAI* 22, 1-2, 9-38.
- Galyardt, A. and Goldin, I. 2015. Move your lamp post: Recent data reflects learner knowledge better than older data. *JEDM*, 7(2), pp.83-108.
- Goldin, I.M., Koedinger, K.R., and Aleven, V. (2013). Hints: You can't have just one. In *EDM 2013*, 232-235.
- Holstein, K., Hong, G., Tegene, M., McLaren, B. M., & Aleven, V. (in press). The classroom as a dashboard: Co-designing wearable cognitive augmentation for K-12 teachers. To appear in *LAK'18*. ACM.
- Käser, T., Koedinger, K., & Gross, M. (2014). Different parameters-same prediction: An analysis of learning curves. In *EDM 2014*.
- Knuth, E. J., Alibali, M. W., McNeil, N. M., Weinberg, A., & Stephens, A. C. 2011. Middle school students' understanding of core algebraic concepts: Equivalence & variable. *Advances in Mathematics Education: Early algebraization*, 259-276, Springer Berlin Heidelberg.
- Koedinger, K. R., Baker, R. S. J. d., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. 2010. A data repository for the EDM community: The PSLC DataShop. *Handbook of EDM*, 43–55, Boca Raton, FL: CRC Press.
- Koedinger, K.R., Corbett, A.T., and Perfetti, C. 2012. The Knowledge-Learning-Instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science* 36, 5, 757–798.
- Koedinger, K. R., McLaughlin, E. A., & Stamper, J. C. 2012. Automated student model improvement. In *EDM 2012*, 17-24.
- Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. 2013. Using data-driven discovery of better student models to improve student learning. In *AIED 2013*, 421-430, Berlin, Heidelberg: Springer.
- Koedinger, K. R., Yudelson, M. V., & Pavlik, P. I. (2016). Testing theories of transfer using error rate learning curves. *TICS*, 8(3), 589-609.
- Kulik, J. A., & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems: a meta-analytic review. *RER*, 86(1), 42-78.
- Li, N., Cohen, W., Koedinger, K., & Matsuda, N. 2011. A machine learning approach for automatic student model discovery. In *EDM*, 31-40.
- Liu, R., Koedinger, K.R., and McLaughlin, E.A. 2014. Interpreting model discovery and testing generalization to a new dataset. In *EDM*, 107-113.
- Liu, R., & Koedinger, K. R. (2017). Closing the loop: Automated data-driven cognitive model discoveries lead to improved instruction and learning gains. *JEDM*, 9(1), 25-41.
- Liu, R. and Koedinger, K.R. 2015. Variations in learning rate: Student classification based on systematic residual error patterns across practice opportunities. In *EDM*, Educational Data Mining Society, Worcester, MA.
- Long, Y., & Aleven, V. 2014. Gamification of joint student/system control over problem selection in a linear equation tutor. In *ITS 2014*, 378-387.
- Long, Y., and Aleven, V. 2017. Educational game and intelligent tutoring system design: A classroom study and comparative design analysis. *ACM ToCHI* 24, 3.
- Long, Y. and Aleven, V. 2017. Enhancing learning outcomes through self-regulated learning support with an open learner model. *UMUAI* 27, 1, 55-88.
- Lovett, M.C. 1998. Cognitive task analysis in the service of intelligent tutoring system design: A case study in statistics. In *ITS 1998*, 234-243.
- Lovett, M., Meyer, O., and Thille, C. 2008. The Open Learning Initiative: Measuring the effectiveness of the OLI statistics course in accelerating student learning. *JIME* 2008, 1.
- Martin, B., Mitrovic, T., Mathan, S., and Koedinger, K.R. 2011. Evaluating and improving adaptive educational systems with learning curves. *UMUAI*, 21, 249–283.
- Mathan, S.A. and Koedinger, K.R. 2005. Fostering the intelligent novice: Learning from errors with metacognitive tutoring. *Educational Psychologist* 40, 4, 257-265.
- Mavrikis, M., Gutierrez-Santos, S., & Poulovassilis, A. (2016). Design and evaluation of teacher assistance tools for exploratory learning environments. In *LAK (2016)*, 168-172. ACM.
- Pavlik P.I., Cen H., Koedinger K.R. 2009. Performance Factors Analysis -- A new alternative to knowledge tracing. In: Dimitrova V., Mizoguchi R., du Boulay B., Graesser A. (eds.). In *AIED 2009*, 531 - 538. Amsterdam: IOS Press
- Romero C, Ventura S, Pechenizkiy M, Baker RS. *Handbook of educational data mining*. CRC Press; 2010
- Spada, H., & McGaw, B. 1985. The assessment of learning effects with linear logistic test models. *Test design: Developments in psychology and psychometrics*, 169-194, Orlando, FL: Academic Press.
- Stamper, J. C., & Koedinger, K. R. 2011. Human-machine student model discovery and improvement using DataShop. In *AIED*, 353-360.
- Stamper, J., Koedinger, K., & McLaughlin, E. (2013). A comparison of model selection metrics in Datasshop. In *EDM 2013*.
- Waalwijk, M., Aleven, V., & Taatgen, N. 2013. Does supporting multiple student strategies lead to greater learning and motivation? Investigating a source of complexity in the architecture of intelligent tutoring systems. *Computers & Education*, 60(1), 159 – 171.
- Wilson, M. and De Boeck, P. 2004. Descriptive and explanatory item response models. In *Explanatory Item Response Models: A Generalized Linear and Nonlinear Approach*, Springer, New York, 43-74.
- Xhakaj, F., Aleven, V., & McLaren, B. M. (2016). How teachers use data to help students learn: Contextual inquiry for the design of a dashboard. In *EC-TEL 2016* (pp. 340-354). Springer International.
- Yudelson, M., & Koedinger, K. (2013). Estimating the benefits of student model improvements on a substantive scale. In *EDM 2013*.